Before and after: "Critical event" analysis with longitudinal data using SAS®

Robert Bauserman, Maryland Medical Research Institute, Baltimore, MD
Doug Thompson, Maryland Medical Research Institute, Baltimore, MD

ABSTRACT

In large samples followed over time, a "critical event" of interest may occur (such as pregnancy or disease diagnosis or movement of a measurement value past a threshold). Analyses of such events, including their predictors, correlates, or sequelae, may require examining variables of interest based on when they occur relative to the critical event. However, the critical event may occur at different timepoints (such as study visits or ages) for different participants. This paper presents code and macros developed for a longitudinal research study that mark the timepoint at which the "critical event" occurs and arrange repeated measurements for each subject in terms of the number of measurements before or after the critical event.

INTRODUCTION

Longitudinal research studies may involve a "critical event" (CE) of interest (such as a developmental milestone, onset of a disease diagnosis, implementation of an intervention where a pre- and post-time series is available, or movement of a measurement value past a threshold). This event may be present for some subjects, but not others, and may take place at different points in time for different subjects. Analyses of critical events, their predictors, and sequelae may require examining specific predictors or outcomes on the basis of when they occur *relative to the CE*. However, because the CE may occur at different times for different participants, this makes it necessary to correctly identify and mark when observations occur relative to the critical event. Only then can later modeling or other analyses make comparisons or predict outcomes at similar time intervals before and after the CE for all subjects.

The code presented here was developed for analysis of data from a longitudinal health and growth study of adolescent girls that included 10 annual visits with anthropometric measurements. The "critical event" of interest was first birth of a child at an early age (defined as births that occurred at or before age 16), and the analysis was intended to compare girls with and without early births on later measures of health-related variables, such as Body Mass Index (BMI) and blood lipids.

The two macros presented here (along with details of preliminary data processing) were designed to transform multiple-observation longitudinal data files into multiple-variable data files that are structured relative to the age or time interval at which the critical event occurred for each subject. We want to move from something like this:

Table 1.
Initial data file (partial listing)

ID	Visit	Critical event occurs (1=yes, 0=no)	Variable 1
001	1	0	2
001	2	0	3
001	3	0	1
001	**4**	**1**	**4**
001	5	0	6
001	6	0	5
002	1	0	1
002	2	0	1
002	**3**	**1**	**2**
Etc.	X	X	X

To this:

Table 2.
Final data file

ID	Var1_pre4	Var1_pre3	Var1_pre2	Var1_pre1	Var1_onset	Var1_post1	Var1_post2	Var1_post3
001	.	2	3	1	**4**	6	5	.
002	.	.	1	1	**2**	2	3	2
003	2	1	3	2	**4**	4	.	.

Note that the values of variable 1 for each subject at the time of the critical event are now aligned in a single new column (Var1_onset), although the CE occurred at different visits for each subject. The values of variable 1 for the visit that immediately preceded the CE are now aligned in the column Var1_pre1 (indicating the first visit prior to the CE), values for the first visit immediately after the CE are now aligned as the column Var1_post1, and so on. Because the number of measurements made both before and after the CE varies from subject to subject, some rows have missing data for visits. With every measurement now aligned in a new variable that indicates the timepoint either before or after the CE at which that value was measured, it is possible to analyze longitudinal changes with reference to total or elapsed time before or after the CE.

PRELIMINARY STEPS

The first segments of code presented here are not part of the macros, but instead represent preliminary processing that we needed to do for the analyses of interest in this study. Some or all of these steps may be irrelevant to other analyses of critical events with different data. In the growth and health study, our purpose was to examine whether experiencing the CE of early birth related to later health measures, so we wanted to use girls who did not experience the CE as a comparison group. But if we are restructuring data relative to the time of the CE, what do we do with participants who did not experience the CE? In the growth and health study, participants were followed throughout adolescence, a period when outcomes of interest (such as BMI) are expected to change dramatically regardless of the occurrence of early pregnancy and birth.

To examine whether or not early birth affected a later outcome, girls without such an event would need to be assigned a dummy value for the CE age, and the distribution of ages for these dummy values in the controls would have to be in the same proportions as the ages of actual CEs in the cases. We used the code shown here to (a) identify participants who did not experience the CE, to serve as a comparison group for outcomes (the “controls”); (b) randomly assign equal numbers of comparison participants to each participant that experienced the CE (the “cases”); (c) assign the controls an “onset age” that matched the age of the CE for the case to which they were assigned; and (d) restructure the multiple-observation dataset into a multiple-variable (one observation per subject) dataset.

These steps ensured that comparisons between girls with the CE (the “cases”) and those without (the “controls”) were based on measurements at similar ages. For example, suppose we wanted to test if early birth had an effect on BMI three years after the pregnancy. Comparison girls had no early births, creating the problem of how to identify an appropriate age for comparison of their BMIs to those of the cases. If 40% of early-birth girls reported the first birth at age 16, 30% at age 15, and 30% at age 14, then randomly assigning equal proportions of control girls to each case and then assigning an “onset age” to match the case’s age at first birth means that 40% of comparison girls would be assigned an onset age of 16, 30% age 15, and 30% age 14. Then, comparisons of an outcome between the cases and controls – such as BMI three years post-CE – would appropriately reflect different ages of occurrence for the CE. In datasets where all subjects experience the CE at some point, or where subjects with no CE are to be dropped from analyses because no comparison group is needed, then preliminary steps (a) through (c) are unnecessary.

(a) Identify participants who did and did not experience the CE. These steps create two files, with only IDs

and CE-related variables, for participants with and without the CE. In the code example below, the original datafile is teenpreg; the preexisting variable that records when the critical event occurred is age_first_birth (a similar variable for any data file with a CE would simply specify the time interval when the event of interest occurred, and would be entered as missing if the CE did not occur); and the newly created dummy variable that flags the occurrence of the CE is gave_birth16.

```
/* Identify subjects WITH critical event (CE) */
/*This creates a separate file with ONLY those girls who reported an early birth*/
data gave_birth16;
        set teenpreg;
        keep id age_first_birth gave_birth16;
                if age_first_birth^=. and age_first_birth<=16 then do;
                        gave_birth16=1;
                        output;
                end;
        label gave_birth16='Gave birth at/before age 16 (=1) vs. did not give birth (=0)';
run;
proc sort data=gave_birth16 out=gave_birth16 nodupkey; by id; run;

/* Identify subjects WITHOUT critical event */
proc sort data=teenpreg out=ids(keep=id age_first_birth) nodupkey; by id; run;

data did_not_give_birth16;
        merge ids(in=in1) gave_birth16(in=in2);
        by id;
                if in1=1 and in2^=1 then do;
                        gave_birth16=0;
                        output;
                end;
run;
```

(b) Randomly assign equal numbers of participants without the CE ("controls") to each participant who did experience the CE ("cases"). These steps simply count girls with and without the CE to identify how many "controls" to assign to each "case", then use the RANUNI function in creating groups of girls that each have one early-birth girl and the appropriate number of comparison girls.

```
/* Count number giving birth before age 16  ('cases') */
data _null_;
        set gave_birth16 end=eof;
        retain number_gave_birth16 0;  /* initializes count to 0 */
        number_gave_birth16+1;         /* increments count by 1 */
                if eof=1 then do;
                  call symput('n_birth',number_gave_birth16);
                end;
run;
%put number giving birth before age 16 = &n_birth;

/* Count number NOT giving birth before age 16 ('controls') */
data _null_;
        set did_not_give_birth16 end=eof;
        retain number_not_gave_birth16 0;
        number_not_gave_birth16+1;
                if eof=1 then do;
                  call symput('n_no_birth',number_not_gave_birth16);
                end;
```

```
run;
%put number NOT giving birth before age 16 = &n_no_birth;
/* Determine number of comparison girls to get as 'controls' for each 'case' */
%let num_controls = %eval(&n_no_birth / &n_birth);
%put number of controls per girl giving birth before age 16 = &num_controls;

/*Randomly identify controls (no births before 16) to match with each case*/
```

Note: a 5-digit random seed number for the RANUNI function was obtained from www.random.org. This web-site creates genuine random numbers from atmospheric noise, rather than using a pseudo-random number generator.

```
data all_controls;
        set did_not_give_birth16;
        rand=ranuni(81070);
        keep id rand;
run;
proc sort data=all_controls out=all_controls; by rand; run;

/* First identify n matched controls, where n = integer part of [(n gave_birth16=0)/(n gave_birth16=1)] */
/* SETNUMBR identifies which matched controls will be linked with which case */
data matched_controls;
        set all_controls;
        retain count setnumbr 0;
        get_num=symget('num_controls');
        n_birth=symget('n_birth');
        tot_get=get_num*n_birth;

        if int(count/get_num) = (count/get_num) then setnumbr+1;
        count+1;
        control=0;
        if _n_<=tot_get then output;
        label control='Matched control (=0) vs. case (=1)';
run;

/* Number the cases */
data gave_birth16;
        set gave_birth16;
        retain setnumbr 0;
        setnumbr+1;
        control=1;
run;

/* Create a combined dataset, with cases and matched controls */
data cases_and_controls;
        set gave_birth16 matched_controls;
        keep id control setnumbr age_first_birth;
run;
proc sort data=cases_and_controls out=cases_and_controls; by setnumbr DESCENDING control; run;
```

(c) Assign controls an "onset age" that matches the age of the CE for the case to which they were assigned.

```
/* Give matched controls the same 'onset age' as the case */
data cases_and_controls;
        set cases_and_controls;
```

```
        retain case_control_age;
        by setnumbr;
        if first.setnumbr=1 then case_control_age=age_first_birth;
        label case_control_age='Critical event age';
run;
```

Now we have a dataset with subsets of equal n that include a single 'case' subject and randomly selected control (comparison) subjects that did not experience the CE. This 'stripped-down' dataset has only subject IDs, set numbers, case-vs.-control status, and a variable indicating the time interval at which the critical event occurred.

Table 3.
Data file with set numbers and assigned ages for controls

ID	Setnumbr	Case_control_age (Visit # of CE (for cases) or assigned "onset age" (for controls))	Control (1 = experienced critical event)
001	1	**4**	**1**
004	1	4	0
005	1	4	0
002	2	**3**	**1**
006	2	3	0
007	2	3	0
003	3	5	1
Etc.	X	X	X

(d) The next step is to create a single-observation-per-subject file from the multiple-observation file. We can use the RETAIN and ARRAY statements and nested DO-loops to create a file with sequential variables containing the measurements of each outcome for each subject (with 10 visits for our study sample, we created 10 sequentially numbered variables for each outcome). In this step, the number of variables created to hold measurements of each outcome is equal to the number of time intervals in the longitudinal dataset. If you had 20 measurements you would create 20 variables for each outcome, and so on.

```
/* Create by-girl dataset (one summary record per respondent) */
proc sort data=teenpreg out=teenpreg; by id visit; run;

data by_girl1;
        set teenpreg;
        retain intage1-intage10 bmi1-bmi10 ave1_1-ave1_10 ... lastvar_1-lastvar_10;
```

Note that integer age (the variable 'intage') was tracked in the NGHS study, is not the same as visit number, and generally increased by 1 with each visit (in some cases, the next annual study visit occurred before the next birthday). For many data sets with critical events, integer age at each visit/time point may not be relevant.

```
        by id visit;
/* you need to create an array for each variable that you will later analyze */
        array intagea {10} intage1-intage10;
        array bmia {10} bmi1-bmi10;
        array ave1_a {10} ave1_1-ave1_10;
        array lastvar_a {10} lastvar_1-lastvar_10;

        if first.id=1 then do;
                do i=1 to 10;
```

```
                    intagea{i}=.;
                    bmia{i}=.;
                    ave1_a{i}=.;
                    lastvar_a{i}=.;
                end;
                drop i;
        end;

        do j=1 to 10;  /* loop through each visit to populate the arrays with data */
          if visit=j then do;
                intagea{j}=intage;
                bmia{j}=bmi;
                ave1_a{j}=ave1;
                lastvar_a{j}=lastvar;
          end;
        drop j;
        end;

        keep id intage1-intage10 bmi1-bmi10 ave1_1-ave1_10 lastvar_1-lastvar_10;
        if last.id=1 then output;   /* */
run;
```

The output data file will contain subject IDs and values of each outcome variable, arrayed in chronological order. However, the case vs. control and set number identifiers are in a different file. These values do not change over time for individual subjects, so we do the following merge:

```
/* Merge the analysis variables with the case/control identifiers */
/* Note: output file 'check' will contain subjects, if any, that appear in the arrayed data file but were not
found in the case-control file */
proc sort data=by_girl1 out=by_girl1 nodupkey; by id; run;
proc sort data=cases_and_controls out=cases_and_controls nodupkey; by id; run;

data cases_and_controls check;
        merge by_girl1(in=in1) cases_and_controls(in=in2);
        by id;
        if in2=1 then output cases_and_controls;
        if in2=1 and in1^=1 then output check;
run;
proc sort data=cases_and_controls out=c_and_c; by setnumbr DESCENDING control; run;
```

All this preliminary work results in a single-observation-per subject dataset that includes separate variables for the measurement of each specific outcome at each visit (e.g., bmi_1-bmi_10), and groups subjects into equal-n subsets with one case and randomly selected comparisons or controls. The data is now structured this way (suffixes indicate sequential time intervals, which can vary from study to study – hours, days, years – as long as they are consistent within the study):

Table 4
Data after conversion to multiple variables per subject (partial list)

ID	Var1_1	Var1_2	Var1_3	Var1_4	Var1_5	Var1_6
001	2	3	1	**4**	6	5
002	1	1	**2**	2	3	2
003	2	1	3	2	**4**	4

Now the macros can be applied.

MACRO #1 (ORDVAR)

The first macro marks time intervals for the critical event (CE), the last measurement before the CE, and the first measurement afterwards. In addition, this macro imputes age-at-visit where it is missing, and handles some special situations that may arise in using integer ages that can be the same at two timepoints. As noted, you should only have to run this macro once for the data file you want to analyze. Note that the code has embedded comments on the function of some sections, in addition to comments added in this paper.

```
/*************************************************************/
/* Macro ORDVAR identifies:                                  */
/* 1) the VISIT of the critical  event (created as variable 'time')  */
/* 2) the first VISIT after (created as variable 'firstpost'), and   */
/* 3) the last VISIT before (created as variable 'lastpre')  */
/*************************************************************/

* YOU SHOULD ONLY NEED TO RUN THIS ONCE;
* Note that algorithms are needed to handle visits with tied INTEGER ages and situations where the girl
was not interviewed at the age of the critical event;
*When there are ties: The LAST tied visit is considered the visit of the critical event;
*When the girl was not interviewed at the age of the critical event: The LAST visit before the critical age
is used;

options mcompilenote=all;
%macro ordvar1(set=,critical_age=);

/* This section imputes age-at-visit for cases where this is missing */
/* In the NGHS data set, measurements began at either age 9 or age 10, and there were 10 total visits */
```

As noted above, the age at each visit/time point may be unnecessary for many data sets; the code could be edited accordingly.

```
data &set;
        set &set;
        array intagea {10} intage1-intage10;

                do i=1 to 10;
                  if age910=1 and intagea{i}=. then do;
                    if i=1 then intagea{i}=9;
                            else if i=2 then intagea{i}=10;
                          /*and so on, up to the last interval*/
                            else if i=10 then intagea{i}=18;
                  end;
                  else if age910=2 and intagea{i}=. then do;
                    if i=1 then intagea{i}=10;
                            else if i=2 then intagea{i}=11;
                          /*and so on, up to the last interval*/
                            else if i=10 then intagea{i}=19;
                  end;
                end;
        drop i;
run;
```

Regardless of the importance of noting age, the variables 'time,' 'lastpre,' and 'firstpost' must be created for processing in the second macro.

```
/* This section identifies starting/ending places in the series*/
data &set;
set &set;
        array ages {10} intage1-intage10;
                foundstart =.;
                lastpre =.;
                firstpost =.;
```

If noting the age is unnecessary, appropriate modification of the next section of code would identify 'time,' 'lastpre' and 'firstpost' in most cases.

```
/* Exact age present in series (i.e., girl was interviewed at critical age)* /
do j=1 to 10;
  if ages{j}= &critical_age then do;
    time =j;
    lastpre =j-1;
    firstpost =j+1;
  end;
end;
```

The following code section covered situations where the CE (early birth) was known to occur at a particular age, but the annual visit for the girl at that age was missing.

```
/* Exact age NOT in series (i.e., girl was not interviewed at critical age)* /
/* If missing, time = last previous visit in series* /
if lastpre =. then do;
  do k=1 to 10 until (foundstart =1);
    if k=1 then do;
        if ages{k} <= &critical_age <= ages{k+1} then do;
        time = k;
        lastpre =k;
        firstpost =k+1;
      foundstart =1;
                end;
```

Note: foundstart = 1 can be regarded as a flag for when the girl was not interviewed at the age when the critical event occurred. It is a good idea to check these cases to make sure they are handled as intended.

```
    if &critical_age < ages{k} then do;
      time = .;
        lastpre =.;
        firstpost =k;
      foundstart =1;
    end;
  end;
  if k>=2 and k<=9 then do;
    if ages{k} <= &critical_age <= ages{k+1} then do;
        time = k;
        lastpre =k-1;
        firstpost =k+1;
      foundstart =1;
   end;
  end;
end;
```

```
end;

drop j k ;
label time = 'VISIT number corresponding to the age of critical event'
lastpre = 'Number of last VISIT before the critical event'
firstpost = 'Number of first VISIT after the critical event';

run;
%mend ordvar1;

options mprint;
%ordvar1(set=cases_and_controls,critical_age=case_control_age);
```

MACRO #2 (ORDVAR2)

The second macro moves the values of the measurements of each outcome at each time-point into a new variable structure that orders them sequentially before and after the onset time of the critical event. The macro includes coding to handle special situations that may arise in the data set (such as when the CE occurs at the time of the first measurement, or the last). Note the explanatory comments embedded in the code as well as those added in this paper.

```
/*****************************************************/
/* Macro ordvar2 codes variables in terms of their   */
/* temporal distance from the critical event.        */
/*****************************************************/
/* YOU NEED TO RUN THE MACRO SEPARATELY FOR EACH VARIABLE THAT YOU WANT
TO CODE RELATIVE TO THE CRITICAL EVENT*/

options mcompilenote=all;
%macro ordvar2(set=,var=,critical_age=,impage=0);

/* Identify pre-values before starting point, post-values after starting point, and value at age-of-onset */
/* Note: Here, pre/post values are years (not visits) pre- and post-onset */
data &set;
set &set;
        array ages {10} intage1-intage10;
        array getvar {10} &var.1 - &var.10;
        array pres {10} &var._pre1 - &var._pre10;
        array posts {10} &var._post1 - &var._post10;
        array presage {10} &var._preage1 - &var._preage10;
        array postsage {10} &var._postage1 - &var._postage10;
```

The number of array elements should be set based as the maximum possible number of measurements made. This should be consistent across variables; even if some outcomes were deliberately measured at only some of the visits, the macro will array the measurements at the apporpriate number of time intervals before or after the CE.

```
        if time^=. then &var._onset =getvar{time};
        &var._onsetage = &critical_age;
        yrsbef=.; yrsaft=.; befdiff=.; specialcase=.;
```

The next sections of code all handle special situations.

```
/* Last pre-observation is first observation, and same age as age-of-onset */
```

```
        if lastpre =1 and ages{1}=&critical_age then do;
          specialcase=1;
          pres{10}=getvar{1};
          if pres{10}^=. then presage{10}=ages{1};
          else presage{10}=.;
        end;
/* Last pre-observation is first observation, and before the age-of-onset */
        else if lastpre =1 and ages{1}<&critical_age then do;
          specialcase=2;
          befdiff=&critical_age-ages{1};
          yrsbef=11-befdiff;
         if yrsbef>0 and yrsbef<11 then do;
           pres{yrsbef}=getvar{1};
                if pres{yrsbef}^=. then presage{yrsbef}=ages{1};
                else presage{yrsbef}=.;
         end;
         else if yrsbef<=0 or yrsbef>=11 then do;
          put 'id=' id ' yrsbef=' yrsbef 'onset=' &critical_age 'age=' ages{1};
         end;
        end;

        if firstpost =10 then do;
          specialcase=3;
          yrsaft=(ages{10}-&critical_age);
          if yrsaft>0 and yrsaft<11 then do;
            posts{yrsaft}=getvar{10};
                if posts{yrsaft}^=. then postsage{yrsaft}=ages{10};
                else postsage{yrsaft}=.;
          end;
          else if yrsaft<=0 or yrsaft>=11 then do;
               put 'id=' id ' yrsaft=' yrsaft ' onset=' &critical_age ' age=' ages{10};
          end;
        end;
/* Age-of-onset is before any observations -- all pre-measures missing */
        if lastpre=. and time=. then do;
          specialcase=4;
        end;
/* Do not try to compute pre-values if it is known that there are none */
        if specialcase^=4 then do;
/* Pre-values */
          if lastpre >1 and lastpre <=10 then do;
            do j=lastpre to 1 by -1;
              if &critical_age=ages{j} then yrsbef=10;
                        else if &critical_age>ages{j} then yrsbef=11-(&critical_age-ages{j});
             if yrsbef>0 and yrsbef<11 then do;
               pres{yrsbef}=getvar{j};
                           if pres{yrsbef}^=. then presage{yrsbef}=ages{j};
                           else presage{yrsbef}=.;
                    end;
                    else if yrsbef<=0 or yrsbef>=11 then do;
               put 'id=' id ' yrsbef=' yrsbef ' onset=' &critical_age;
             end;
            end;
           end;
        end;
```

```
/* Post-values */
        if firstpost >=1 and firstpost <11 then do;

          do m=firstpost to 10;
            yrsaft=ages{m}-&critical_age;
            if yrsaft>0 and yrsaft<11 then do;
              posts{yrsaft}=getvar{m};
                        if posts{yrsaft}^=. then postsage{yrsaft}=ages{m};
                        else postsage{yrsaft}=.;
                end;
                else if yrsaft<=0 or yrsaft>=11 then do;
             put 'id=' id ' yrsaft=' yrsaft ' onset=' &critical_age;
           end;
          end;
        end;
        drop j m;
        run;

%if &impage=1 %then %do;
```

The next sections were used to impute missing integer age values in the NGHS data set; as noted earlier, this may be unnecessary in many cases where subject age at each time point is not a concern.

```
data &set;
        set &set;

        array presage {10} &var._preage1 - &var._preage10;
        array postsage {10} &var._postage1 - &var._postage10;

        if &critical_age^=. then do;
          do i=1 to &critical_age;
            currpre=11-i;
                currpreage=&critical_age-i;
                if presage{currpre}=. then do;
              presage{currpre}=currpreage;
            end;
          end;
          do j=1 to 10;
            currpost=j;
                currpostage=&critical_age+j;
                if postsage{currpost}=. then do;
                  postsage{currpost}=currpostage;
                end;
          end;
        end;
        drop i j;
        run;

%end;
%mend ordvar2;
```

As noted above, you need to run the macro separately for each variable to be coded relative to onset age. For example, to call the macro to process the variable bmi (Body Mass Index):

```
*ordvar2 for Body Mass Index (bmi);
```

```
options mprint;
%ordvar2a(set=cases_and_controls,var=bmi,critical_age=case_control_age,impage=1);
```

Now you have each measurement of the outcome variable coded in terms of time intervals before the CE (here, bmipre1-bmipre10, with the higher numbers indicating more intervals prior to the CE); the time interval of the CE itself (bmi_onset); and time intervals after the CE (here, bmi_post1-bmi_post5, with the higher numbers indicating more elapsed time after the CE).

Table 5.
Structure of final data file (partial listing)

ID	Var1_pre 4	Var1_pre 3	Var1_pre 2	Var1_pre 1	Var1_ons et	Var1_pos t1	Var1_pos t2	Var1_pos t3
001	.	2	3	1	**4**	6	5	.
002	.	.	1	1	**2**	2	3	2
003	2	1	3	2	**4**	4	.	.

You can now proceed to any desired time-based analyses with these variables, including descriptive analyses (e.g., figures showing each measure at the age of onset, before and after), mixed models or time series models.You can even transpose back to a multiple-observations-per-subject dataset, if this is convenient for some analyses (e.g., PROC MIXED or GEE using GENMOD).

CONCLUSIONS

The macros presented here are a means of restructuring a longitudinal data file so that it is possible to look at the impact of a critical event on an outcome variable, even if systematic change in the outcome is expected over time regardless of the occurrence of the critical event. Although originally created for a study on health and growth in adolescence, the macros could be modified to process any longitudinal data set with a similar structure.

ACKNOWLEDGMENTS

SAS is a Registered Trademark of the SAS Institute, Inc. of Cary, North Carolina.

CONTACT INFORMATION

YOUR COMMENTS AND QUESTIONS ARE VALUED AND ENCOURAGED. CONTACT THE FIRST AUTHOR AT:

Robert Bauserman, PhD
Research Analyst
Maryland Medical Research Institute (MMRI)
600 Wyndhurst Ave.
Baltimore, MD 21210
Work phone: (410) 532-0249
Email: **rbauserman@mmri.org**
Web: www.mmri.org

Doug Thompson, PhD is a consultant with MMRI and can be reached at dthompson@mmri.org.